

|                             |      |
|-----------------------------|------|
| Assembler/Editor cold start | 3743 |
| Assembler warm start        | 3782 |
| Input a character           | 36BD |
| Output a character          | 36D3 |
| Carriage return             | 36D1 |
| Bell                        | 376C |
| Exit to APPLE monitor       | 377C |

The last memory location actually used by the ARESCO Assembler/Text Editor is \$37A2. Within the assembler/editor, all character data is in true ASCII code, not the quasi-ASCII normally used for APPLE I/O.

#### Source Text Format

Source Text is stored in APPLE memory in standard ASCII. Each line of text stored in memory begins with the two byte line number in BCD, and is stored in high-order, low-order byte sequence; not as low, high. Each line is terminated with a 0D (carriage return). The end-of-file mark is 1F following the final 0D.

#### For More Information

A complete, commented listing of the original ARESCO Assembler/Text Editor for 6502 based microsystems is available from ARESCO for \$20.00. The original version does not contain some of the minor patches which have been added to the package over the past three years.

ARESCO also publishes the Rainbow, a newsletter devoted exclusively to APPLE II owners. See the last page of this document for subscription information.

APPENDIX B  
Sample Runs

A disassembled listing of a subroutine to "click" the APPLE's speakers was published in the ARESCO Rainbow in January, 1979. (See Basic Music & Sound Effects, Issue 1, Volume 1, page 17.)

A greater segment of the program from which that subroutine was taken is used here to illustrate the operation of the ARESCO Assembler/Text Editor Package For The APPLE II.

First, the Editor portion of the package is entered and the program is listed. The assembler is called, and the program is listed during assembly. The asterisks in lines 110, 120, and 130 indicate the space reserved for forward references. In the later disassembly, the addresses referenced are present, followed by an EA (NOP) instruction.

After the modifications for the two-pass version of the assembler have been made, the editor is again entered, and the assembler is called. The asterisks are still present, and spurious errors have been generated.

The errors are ignored, and the assembler is entered at \$3782 for the second pass. Now the forward references are properly calculated and the asterisks replaced. Again, the resulting machine code is shown, then disassembled using the APPLE's resident disassembler. Notice the absence of the EA instruction in the branch instructions in the two-pass assembler version.

Sample Editor Run

|                |                       |
|----------------|-----------------------|
| *37436         | Start the Editor      |
| BASE=4000      | Text starts at \$4000 |
| N OR 0?0       | Specify old file      |
| 4000 4147 0018 | Status                |
| P.B            | Print out the file    |

```

0010 ;SUBROUTINE TO GENERATE TONES
0020 *=$000
0030 PITCH *==+1
0040 LONGL *==+1
0050 CLICK = $C030 ; CLICK SPEAKER
0060 WAIT = $FC08 ; DELAY SUB.
0070 ENTRY LDA CLICK ; MAKE CLICK
0080 LDA PITCH
0090 JSR WAIT
0100 LDA LONGL
0110 BNE LOOP
0120 DEC LONGL
0130 BEQ END
0140 LOOP DEC LONGL
0150 JMP ENTRY
0160 END RTS
0170 LONGL , BYTE 00 ; DURATION HI BYTE
0180 , END
*ET

```

A

Ask for assembly  
Single-pass version

PRESO

LINE # LOC CODE LINE

```

0010 0200      ; SUBROUTINE TO GENERATE TONES
0020 0200      *$000      Set origin
0030 0000      PITCH    **++1      Reserve space
0040 0001      LONGL    **++1
0050 0002      CLICK    = $0030      ; CLICK SPEAKER
0060 0002      WAIT     = $FC00      ; DELAY SUB.      Define subroutines
0070 0002 AD 30 C0 ENTRY LDA CLICK      ; MAKE CLICK
0080 0005 AD 00 00      LDA PITCH
0090 0008 20 A0 FC      JSR WAIT
0100 0008 AD 01 00      LDA LONGL
0110 000E D0 ** **      BNE LOOP
0120 0011 CE ** **      DEC LONGH      Forward references
0130 0014 F0 ** **      BEQ END
0140 0017 CE 01 00 LOOP DEC LONGL
0150 001A 4C 02 00      JMP ENTRY
0160 001D 00      END      RTS
0170 001E 00      LONGH    .BYTE 00      ; DURATION HI BYTE
0180 001F      .END

```

ERRORS = 0000

## SYMBOL TABLE

|       |      |       |      |       |      |      |      |
|-------|------|-------|------|-------|------|------|------|
| PITCH | 0000 | LONGL | 0001 | CLICK | 0030 | WAIT | FC00 |
| ENTRY | 0002 | LOOP  | 0017 | LONGH | 001E | END  | 001D |

END OF ASSEMBLY

#000, 81F

Show results

```

0000- 00 00 AD 30 C0 AD 00 00
0005- 20 A8 FC AD 01 08 D0 07
0010- EA CE 1E 08 F0 07 EA CE
0015- 01 08 4C 02 08 60 00 A8

```

#000, 81FLDisassemble the code  
with the APPLE monitor

```

0000- 00      BRK
0001- 00      BRK
0002- AD 30 C0  LDA  $C030
0005- AD 00 00  LDA  $0000
0008- 20 A8 FC  JSR  $FCA8
000B- AD 01 08  LDA  $0001
000E- D0 07    BNE  $0017
0010- EA      NOP
0011- CE 1E 08  DEC  $001E
0014- F0 07    BEQ  $001D
0016- EA      NOP
0017- CE 01 08  DEC  $0001
001A- 4C 02 08  JMP  $0002
001D- 60      RTS
001E- 00      BRK

```

Generated by forward references  
in branch instructions

# Sample Run Using The Two-Pass Modification To The Assembler

\*37436

Enter The Editor

BASE=4000

N OR 0?0

4000 4147 0018

A

Assemble pass one

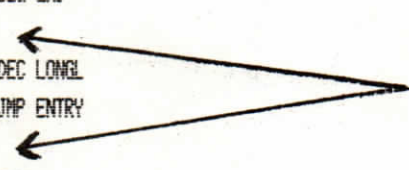
PRESO

LINE # LOC CODE LINE

```

0010 0200      ;SUBROUTINE TO GENERATE TONES
0020 0200      *=$000
0030 0000      PITCH  **++1
0040 0001      LONGL  **++1
0050 0002      CLICK  = $C030      ; CLICK SPEAKER
0060 0002      WAIT   = $FC08      ; DELAY SUB.
0070 0002 AD 30 C0 ENTRY LDA CLICK      ; MAKE CLICK
0080 0005 AD 00 08      LDA PITCH
0090 0008 20 A8 FC      JSR WAIT
0100 0008 AD 01 08      LDA LONGL
0110 000E ** **      BNE LOOP
0120 0010 CE ** **      DEC LONGL
0130 0013 ** **      BEQ END
***ERROR # 04 PC = CEFF
0140 0015 CE 01 08 LOOP DEC LONGL
0150 0018 4C 02 08      JMP ENTRY
***ERROR # 04 PC = CEFF
0160 001B 00      END      RTS
0170 001C 00      LONGH    .BYTE 00      ; DURATION HI BYTE
0180 001D      .END

```



Ignore the errors

ERRORS = 0002

## SYMBOL TABLE

|       |      |       |      |       |      |      |      |
|-------|------|-------|------|-------|------|------|------|
| PITCH | 0000 | LONGL | 0001 | CLICK | 0030 | WAIT | FC00 |
| ENTRY | 0002 | LOOP  | 0015 | LONGH | 001C | END  | 001B |

END OF ASSEMBLY

\*3782G

Start pass two

PRESO

LINE # LOC CODE LINE

```

0010 001D      ; SUBROUTINE TO GENERATE TONES
0020 001D      *=$000
0030 0000      PITCH  *++1
0040 0001      LONGL  *++1
0050 0002      CLICK  = $0030      ; CLICK SPEAKER
0060 0002      WAIT   = $FC00      ; DELAY SUB.
0070 0002 AD 30 C0 ENTRY LDA CLICK      ; MAKE CLICK
0080 0005 AD 00 00      LDA PITCH
0090 0005 20 A0 FC      JSR WAIT
0100 0006 AD 01 00      LDA LONGL
0110 000E D0 05        BNE LOOP
0120 0010 CE 1C 00      DEC LONGH
0130 0013 F0 06        BEQ END
0140 0015 CE 01 00 LOOP DEC LONGL
0150 0018 4C 02 00      JMP ENTRY
0160 001B 60          END   RTS
0170 001C 00          LONGH .BYTE 00      ; DURATION HI BYTE
0180 001D          .END

```

Forward references OK

ERRORS = 0002

Note that error count is  
not reset after pass one

## SYMBOL TABLE

|       |      |       |      |       |      |      |      |
|-------|------|-------|------|-------|------|------|------|
| PITCH | 0000 | LONG1 | 0001 | CLICK | 0030 | WAIT | FC08 |
| ENTRY | 0002 | LOOP  | 0015 | LONGH | 001C | END  | 001B |

END OF ASSEMBLY

\*000.81E

Show results

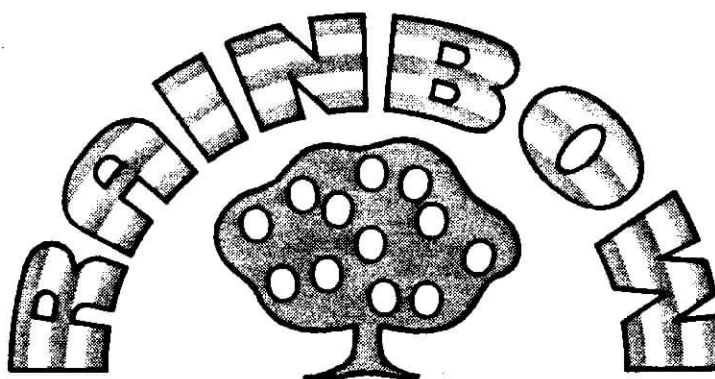
```
0000- 00 00 AD 30 C0 AD 00 00
0000- 20 A8 FC AD 01 03 D0 05
0010- CE 1C 08 F0 06 CE 01 00
0010- 4C 02 00 60 00 60 00
```

\*000.LDisassemble the code  
with the APPLE monitor

```
0000- 00      BRK
0001- 00      BRK
0002- AD 30 C0  LDA $C030
0005- AD 00 00  LDA $0000
0008- 20 A8 FC  JSR $FC08
000B- AD 01 00  LDA $0001
000E- D0 05     BNE $0015
0010- CE 1C 08  DEC $001C
0013- F0 06     BEQ $001B
0015- CE 01 00  DEC $0001
0018- 4C 02 00  JMP $0002
001B- 60      RTS
001C- 00      BRK
```

Note that error count is  
not reset after pass one





**BOX 43, AUDUBON, PA 19407**

(215) 631-9052

The Rainbow is an independent, national newsletter dedicated to APPLE II owners. It's published ten times a year (every month except July and December) and a subscription includes all ten issues of the current volume.

The Rainbow is a user newsletter dedicated to the non-professional computerist. It's the only newsletter to acknowledge the fact that people have to start somewhere before they get to be "experts".

The Rainbow is intended to be an information exchange - you can learn about the projects for which other people use their APPLES - and share your own experiences and discoveries. Find other APPLE owners in your area - or communicate with APPLE owners in other countries. The whole world of APPLE users is available to you through the pages of the Rainbow!

Send us your name, address, and a check or money order for \$15.00 (the price includes all ten issues of the current volume). By return mail, you'll receive your copy of the Rainbow Questionnaire to fill out and return, so we'll know what you want to see in the newsletter. If you live outside the USA, and don't want to wait the three-to-four weeks required by surface (international) mail, include \$10.00 for airmail postage. Please send us a street address - if we're shipping back issues, we ship via UPS - or an additional \$0.40 per copy for domestic postage. No invoicing. Sorry.

Please ship me all the back issues of this volume, and enter my subscription for the current volume of The Rainbow. I enclose payment in full, including postage if required.

NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY, STATE, ZIP \_\_\_\_\_

CHECK \_\_\_\_\_/MC/VISA CARD # \_\_\_\_\_

If using Master Charge, we need the other 4 digits on the card

EXP DATE \_\_\_\_\_ SIGN \_\_\_\_\_

Mail to: THE RAINBOW \* P.O. BOX 43 \* AUDUBON \* PA \* 19407





